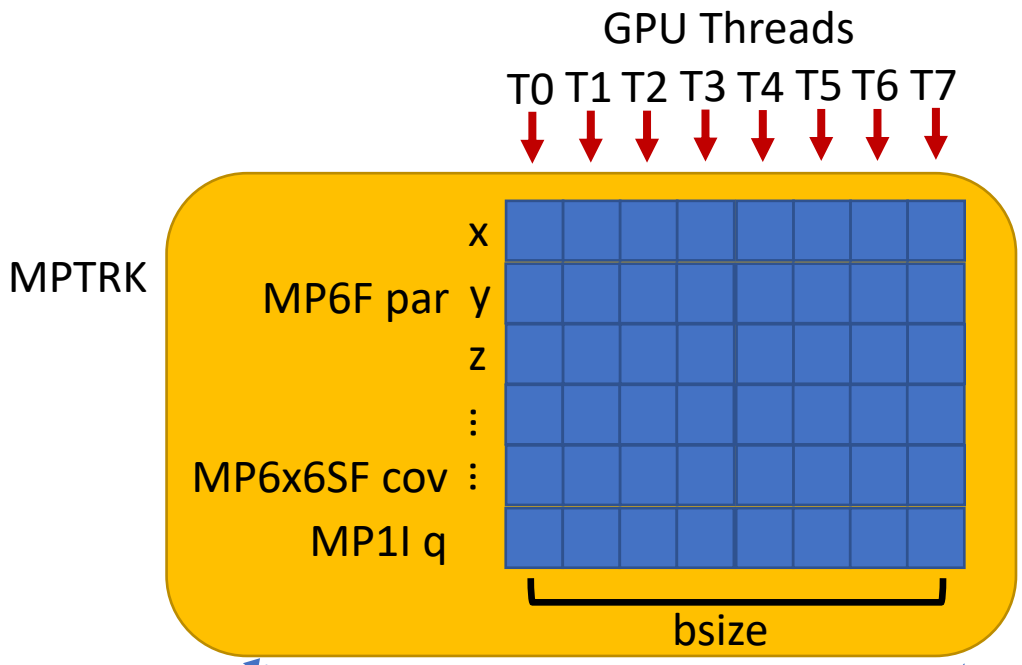
Memory Layouts of the P2Z Data Structures

Seyong Lee

Oak Ridge National Laboratory

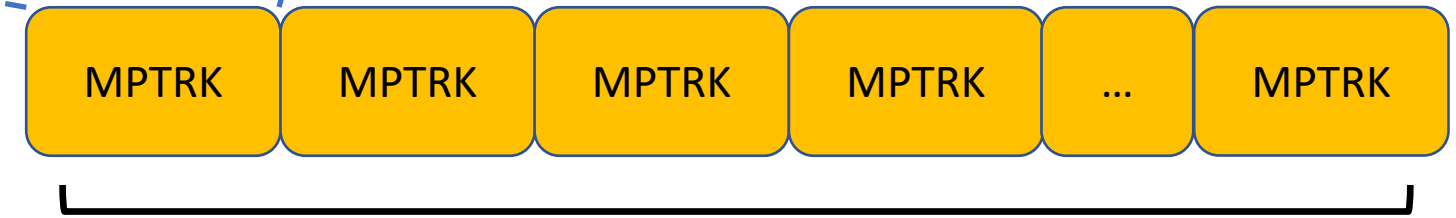
January 12, 2023

Memory Layout of P2Z Data Structures



(1D thread blocks, each of which has *bsize* threads*)
(1D grid of thread blocks)
* For coalesced memory accesses, bsize should be the same as or multiple of the thread block size.

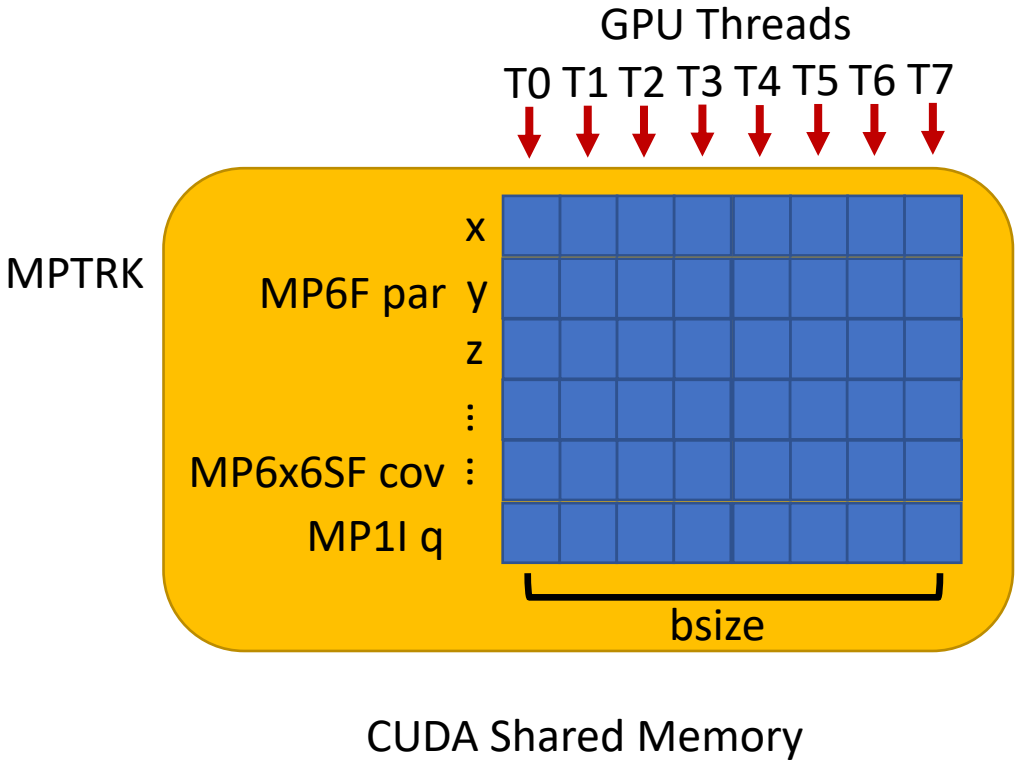
Input *trk* data = an array of *struct MPTRK*



of *struct MPTRK* = ntrks * nevts / bsize

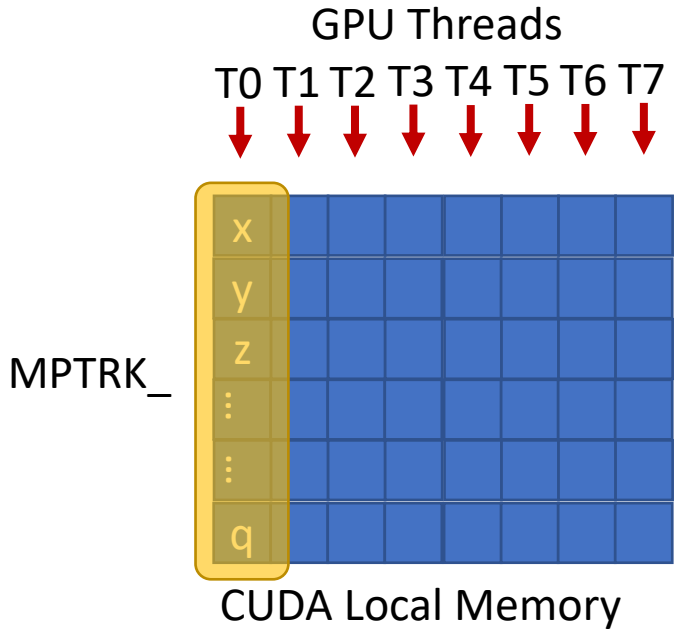
Temporary Data Allocation Strategy

- Option 1: allocate one MPTRK struct on CUDA shared memory per thread block, which is shared by threads in the same thread block.
  - No actual data sharing among threads; each thread accesses different parts of the shared memory.
- Pro:
  - Minimize memory access latency (each thread accesses an element per clock)
- Cons:
  - Too much shared memory usage may limit the number of active warps.
  - Require extra synchronizations to coordinate shared memory accesses among threads.

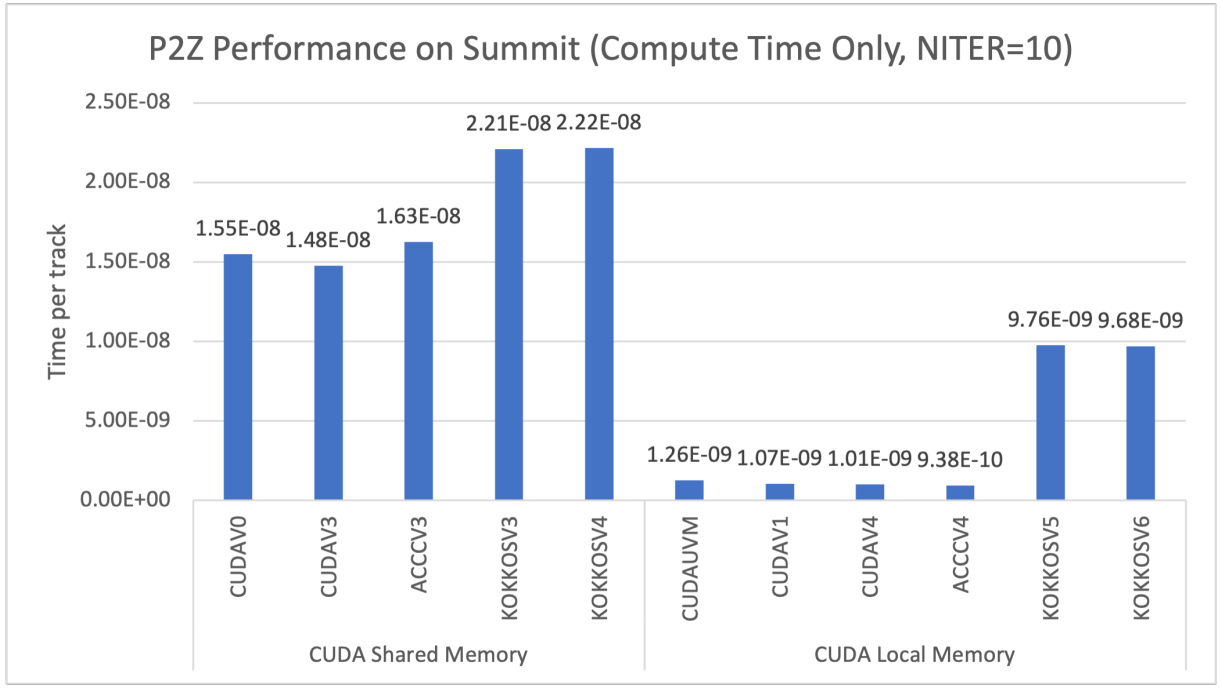


Temporary Data Allocation Strategy (Cont.)

- Option 2: each thread allocates only its used portion of a MPTRK struct on the CUDA local memory (thread-private memory).
- Pro:
  - Minimize CUDA shared memory usage, which may increase the number of active warps.
  - Access to the CUDA local memory is likely to be coalesced.
  - The compiler may allocate some thread-private data on the registers.
- Cons:
  - CUDA local memory uses the same physical memory as CUDA global memory, incurring same access latency as the global memory.



P2Z GPU Performance on Summit (V100)



- **CUDAV0**: CUDA on Unified Memory, Async (10 streams), Shared Memory
- **CUDAV3:** CUDA, Async (10 streams), Shared Memory
- **ACCCV3**: OpenACC C, Async (10 streams), Shared Memory
- **KOKKOSV3**: KOKKOS, Single Async Device Instance, Shared Memory
- **KOKKOSV4**: KOKKOS/CUDA, 10 Async Device Instances, Shared Memory

Compilers: NVCC V11.0, NVHPC V22.11, OpenARC V0.73

- **CUDAUVM**: CUDA C++ on Unified Memory, Async (1 stream), Local Memory
- **CUDAV1:** CUDA on Unified Memory, Async (10 streams), Local Memory
- **CUDAV4**: CUDA, Async (10 streams), Local Memory
- **ACCCV4**: OpenACC C, Async (10 streams), Local Memory
- **KOKKOSV5**: KOKKOS, Single Async Device Instance, Local Memory
- **KOKKOSV6**: KOKKOS/CUDA, 10 Async Device Instances, Local Memory